

Reference 2

Japanese Patent Application Public-disclosure No. 53-63820

Japanese Patent Application Public-disclosure date: June 7, 1978

Title of the invention: Data processing method using logical address/real address translation

Japanese Patent Application No. 51-138768

Japanese Patent Application date: November 18, 1976

[Detailed description of the invention]

The present invention is directed to a data processing method using logical address/real address translation, and in particular to a data processing method for, for example, emulating a target machine on a host machine having a logical address larger than that of the target machine, wherein when the host machine utilizes a TLB to perform logical address/real address translation processing, it uses only a limited storage area in the TLB to thereby, for example, simplify processing for invalidating content of the TLB.

Although the present invention is not limited to the above, for such processing as emulating a target machine on a host machine, a logical address of the host machine is, in general, smaller than that of the target machine and therefore, a predetermined number of zeros is added to the most significant bit of the original logical address to associate the logical address of the target machine, that is, the original logical address with the logical address of the host machine.

Although the original logical address can be easily associated with a logical address of a host machine by adding zeroes in the aforementioned fashion, it causes the following problem. A host machine normally employs a virtual memory system, which frequently issues an instruction to invalidate the content of the TLB (table showing correspondences between logical page addresses and real page addresses), and every time an instruction is issued, the content of the TLB is invalidated

in such a manner that the addresses in the TLB are accessed for invalidation one by one in sequence, and overheads for invalidation processing substantially increase.

Thus, it is an object of the present invention to obviate the aforementioned problem of prior art. It is a further object of the present invention to utilize a limited storage area in a TLB to perform translation processing so that only the limited storage area needs to be invalidated at the time of invalidation processing. To achieve those objects, according to a data processing method using logical address/real address translation of the present invention, in a data processing system which has a logical address greater than a provided original logical address and which performs processing while carrying out logical address/real address translation processing in accordance with a correspondence table establishing correspondences between logical page addresses and real page addresses, the original logical address is associated with the logical address to which a predetermined number of zero bits is added in such a manner that one or more than one zero bit to be added to the logical address exists in the correspondence table access bit region used for accessing the correspondence table on the basis of the logical address, and when performing processing provided by the original logical address, logical address/real address translation processing is conducted by means of the limited storage area in the correspondence table. Hereafter, the present invention will be specifically explained with reference to the attached drawings.

Fig. 1 is a schematic diagram for explaining a conventional data processing method, whereas Fig. 2 is a schematic diagram for explaining a data processing method of the present invention.

In Fig. 1, the numeral 1 denotes an original logical address; 2 a logical address; 3 a correspondence table (TLB) showing correspondences between logical page addresses and real page addresses; 4 a page number comparison means; 5 a translated

real address; 6 a TLB access address; 7 zero bits to be added; and 8 a TLB access bit region.

For example, when associating a logical address of 19 bits from bit A0 to bit A18 with a logical address of 24 bits to perform processing, as is indicated in Fig. 1, zeros equivalent to five bits, i.e., "00000" are added to the most significant bit of the original logical address to determine the logical address 2, so that the host machine performs processing in accordance with the logical address 2.

When the host machine carries out processing in accordance with the logical address 2, the host machine accesses the TLB 3 by means of the TLB access address 6 comprised of, for example, 5th ~ 7th bits and 9th ~ 12th bits of the logical address. In other words, when the logical address 2 is translated to the real address 5 for the host machine to access a storage device (which is not indicated in the drawing), the host machine accesses the TLB 3.

In the address designated by the TLB access address 6 in the TLB 3, a page number (logical page address) corresponding to 0th ~ 4th bits and 8th bit of the logical address, which is paired with a real page address corresponding to the page number, is stored. The pair will be referred to as page address correspondence below.

The host machine determines the real address 5 corresponding to the logical address 2 in the following manner.

- 1) The TLB 3 is accessed by the TLB access address 6 consisting of bits of the provided logical address 2.
- 2) Thereby, page address correspondence is read from the TLB 3 as mentioned above.
- 3) The page number in the read page address correspondence is guided to the page number comparator 4 where it is compared with a page number (consisting of 0th ~ 4th bits and 8th bit of the logical address 2) comprised of the above provided logical address 2.

If it transpires that the page numbers do not match, the comparator 4 issues a mismatch output, which means either that

a page address correspondence corresponding to the page number containing the provided logical address 2 does not exist in the TLB 3 or that, even though it exists, a valid bit V in the page address correspondence is in an invalid state.

4) When a mismatch output is issued, a valid page address correspondence is transferred from a main storage device (not indicated in the drawing) into the TLB 3.

5) On the contrary, when the comparator 4 issues a match output, it means that the page address correspondence read as a result of the processing in the above 2) is valid. Thus, a real page address of the read page address correspondence is set in the higher order bits of the real address 5. In the lower order bits of the real address 5 is set a page address (13th bit ~ 23rd bit) of the provided logical address 2.

6) The storage device (not indicated in the drawing) is accessed by the real address 5 obtained by the above processing 5)).

When the host machine translates the provided logical address to a real address, the above-described processing is performed. Especially when performing emulation operations, the content of the TLB 3 is frequently invalidated in the midst of the emulation processing, and in the subsequent processing page address correspondence is re-transferred from the main storage device (not indicated in the drawing) into the TLB. In performing the aforementioned invalidating processing, valid bits V in all the page address correspondences in the TLB 3 are made invalid at the time of processing TLB clear instruction, and when the page address correspondence is accessed later, the comparator 4 issues a mismatch output and a new page address correspondence is brought onto the TLB. In this case, the host machine cannot perform new processing until valid bits V in all the page address correspondences in the TLB 3 are invalidated and therefore, it is desirable that the invalidating processing be performed at high speed. However, in performing the invalidating processing, the host machine accesses the

addresses in the TLB 3 one by one to thereby invalidate the valid bits V one by one.

Referring to Fig. 1 in consideration of the above, the TLB access address 6 corresponds to the bits A0, A1, A2, A3, A4, A5, A6 and A7 of the original logical address 1, from which it is clear that it corresponds to all the addresses (0th address ~ 127th address) in the TLB 3. Thus, to invalidate the content of the TLB 3, all the addresses in the TLB 3 have to be accessed, despite which a page number to be stored in the TLB 3 corresponds to the 0th bit ~ 4th bit and 8th bit of the logical address 2. Therefore, it becomes clear that only the 8th bit of the bits representing a page number has a valid content and all the other bits have "0". In other words, although only two page address correspondences exist in the TLB 3, the 0th ~ 127th addresses in the TLB 3 have to be accessed in sequence when performing the aforementioned invalidating processing.

Fig. 2 is a schematic diagram indicating an embodiment of the present invention, which obviates the aforementioned problem. In the drawing, the numerals 1 ~ 8 denote the same components as those in Fig. 1 and the numeral 9 denotes a limited storage area in the TLB 3.

When associating the original logical address 1 with the logical address 2 in the present invention, the zero bits 7 to be added are placed in the TLB access bit region 8. In the case shown in Fig. 2, bits A0 ~ A3 in the original logical address 1 are associated with the 9th ~ 12th bits in the logical address 2, bits A4 ~ A7 are associated with 0th ~ 3rd bits in the logical address 2 and the zero bits 7 to be added are associated with the 4th ~ 8th bits in the logical address 2. The aforementioned processing can be easily performed if it is designed in such a manner as to be executed by a micro-program.

Therefore, the TLB access address 6 takes a form of "000A0, A1, A2, A3". In other words, it is regarded that the storage area in the TLB 3 spans only from 0th address to 15th address.

The mode for the host machine to perform processing for

determining the real address 5 based on the associated logical address 2 may be exactly the same as that indicated in Fig. 1. However, to invalidate the content of the TLB 3, only the 0th ~ 15th addresses in the TLB 3 have to be accessed.

As is explained above, when carrying out the above-described emulation processing in accordance with the present invention, it looks as if the TLB 3 has only a limited storage area for its addresses and therefore, processing for invalidating the content of the TLB 3 becomes significantly simpler. Further, as the storage area 9 in the TLB 3 to be used to correspond to the original logical address 1 is restricted to a specified range, processing for multiple original logical addresses can be carried out concurrently while being made independent of one another on the TLB 3.

Although the above embodiment concerns emulation operations, the present invention is not limited to the same.

[Brief explanation of the drawings]

Fig. 1 is a schematic diagram illustrating a prior art data processing method.

Fig. 2 is a schematic diagram illustrating a data processing method of the present invention.

1: original logical address, 2: logical address, 3: TLB, 4: page number comparator, 5: real address, 6: TLB access address, 7: zero bits to be added, 8: TLB access bit region, 9: limited storage region

⑯日本国特許庁
公開特許公報

①特許出願公開
昭53-63820

§Int. Cl.²
G 11 C 8/00
G 06 F 9/00
G 06 F 9/20

識別記号

⑤日本分類
97(7) C 0
97(7) F 0
97(7) F 112

庁内整理番号
6453-56
6745-56
6745-56

⑥公開 昭和53年(1978)6月7日
発明の数 1
審査請求 未請求

(全5頁)

⑦論理アドレス・実アドレス変換を用いるデータ処理方法

川崎市中原区上小田中1015番地
富士通株式会社内

⑧特許出願 昭51-138768

⑨出願人 富士通株式会社

⑩明細書 昭51(1976)11月18日

川崎市中原区上小田中1015番地

⑪発明者 畑中一成

⑫代理人 弁理士 森田寛

明細書

1. 発明の名称 論理アドレス・実アドレス変換を用いるデータ処理方法

徴とする論理アドレス・実アドレス変換を用いるデータ処理方法。

2. 特許請求の範囲

(1) 与えられた元論理アドレスにくらべて大きい論理アドレスをもちかつ論理ページ・アドレスと実ページ・アドレスとの対応表にしたがって論理アドレス・実アドレス変換処理を行ないつつ処理を実行するデータ処理システムにおいて、上記元論理アドレスに所定個数の零ビットを附加した上記論理アドレスを対応せしめるに当って、該論理アドレスにもとづいて上記対応表をアクセスするためには用いる対応表アクセス・ビット領域内に上記附加されるべき零ビットが1個または複数個位置するよう対応せしめ、上記論理アドレスで与えられる処理を実行するに当って、上記対応表内の制限された記憶領域を利用して論理アドレス・実アドレス変換処理を行なうようにしたことを特

(2) 上記対応表の内容を無効化する処理に当って、上記制限された記憶領域に対して無効化処理を実行することを特徴とする特許請求の範囲(1)項記載の論理アドレス・実アドレス変換を用いるデータ処理方法。

(3) 上記元論理アドレスと論理アドレスとの対応づけは、プログラム処理によって行なわれることを特徴とする特許請求の範囲(1)項または(2)項記載の論理アドレス・実アドレス変換を用いるデータ処理方法。

(4) 上記元論理アドレスで与えられる処理はターゲット・マシンに対応した処理であり、上記データ処理システムはホスト・マシンであって、該ホスト・マシンが上記ターゲット・マシンをエミュレートして処理を実行することを特徴とする特許請求の範囲(1)項ないし(3)項のいずれか1つ記載の論理アドレス・実アドレス変換を用いるデータ処理方法。

3. 発明の詳細な説明

本発明は、論理アドレス・実アドレス変換を用いるデータ処理方法、特に例えばターゲット・マシンがもつ論理アドレスにくらべて大きい論理アドレスをもつホスト・マシンによって上記ターゲット・マシンをエミュレートする如きデータ処理方法において、ホスト・マシンがTLBを利用して論理アドレス・実アドレス変換処理を行なうに当つて、上記TLB内の制限された記憶領域を利用するようにし、例えばTLBの内容を無効化する処理を簡易化できるようにした論理アドレス・実アドレス変換を用いるデータ処理方法に関するものである。

本発明はそれに限られるものではないが、ターゲット・マシンをホスト・マシンによってエミュレートする如き処理の場合、一般にターゲット・マシンの論理アドレスにくらべてホスト・マシンの論理アドレスが小さい。このため、上記ターゲット・マシンの論理アドレス即ち元論理アドレスをホスト・マシンの論理アドレスに対応づけるべ

く、上記元論理アドレスの最上位ビットに所定個数の零を附加するようにされる。

この場合、上記元論理アドレスと論理アドレスとの対応づけは簡単であるが、次の如き問題点が存在する。即ち、上記ホスト・マシンは一般に仮想記憶方式を採用しているが、上記エミュレーション処理時にTLB(論理ページ・アドレスと実ページ・アドレスとの対応表)の内容を頻繁に無効化する命令が出る。その都度TLBの内容は無効化されることになるが、一般にTLBの各アドレスを順にアクセスしつつ無効化されることになるために無効化処理のためのオーバ・ヘッドが大となる。

本発明は、上記の如き問題点を生じないようにすることを目的としており、TLB内の制限された記憶領域を利用して変換処理を行なわせておき、無効化処理時には上記制限された記憶領域を無効化すれば足りるようになることを目的としている。そしてそのため、本発明の論理アドレス・実アドレス変換を用いるデータ処理方法は与えられた元

論理アドレスにくらべて大きい論理アドレスをもつた論理ページ・アドレスと実ページ・アドレスとの対応表にしたがつて論理アドレス・実アドレス変換処理を行ないつつ処理を実行するデータ処理システムにおいて、上記元論理アドレスに所定個数の零ビットを附加した上記論理アドレスを対応せしめるに当つて、該論理アドレスにもとづいて上記対応表をアクセスするために用いる対応表アクセス・ビット領域内に上記附加されるべき零ビットが1個または複数個位置するよう対応せしめ、上記元論理アドレスで与えられる処理を実行するに当つて、上記対応表内の制限された記憶領域を利用して論理アドレス・実アドレス変換処理を行なうようにしたことを特徴としている。以下図面を参照しつつ説明する。

オ1図は従来のデータ処理方法を説明する説明図、オ2図は本発明のデータ処理方法を説明する説明図を示す。

オ1図において、1は元論理アドレス、2は論理アドレス、3は論理ページ・アドレスと実ペー

ジ・アドレスとの対応表(TLB)、4はページ・ナンバ比較部、5は変換された実アドレス、6はTLBアクセス・アドレス、7は附加される零ビット、8はTLBアクセス・ビット領域を表わしている。

例えばビットA0ないしA18で与えられる19ビットの論理アドレスを24ビットで構成される論理アドレスに対応づけて処理を実行する場合、従来オ1図の如く、元論理アドレス1の上位に5ビット分の零即ち「00000」を附加して論理アドレス2を決定し、ホスト・マシンは該論理アドレス2にしたがつて処理を実行するようになる。

ホスト・マシンが上記論理アドレス2にしたがつて処理を実行する場合、ホスト・マシンは例えば論理アドレスや5ないし7ビットとオ9ビットないしオ12ビットとによってTLBアクセス・アドレス6が形成され、TLB3をアクセスするようになる。即ち論理アドレス2を実アドレス5に変換して表示しない記憶装置をアクセスする

に当ってTLB3をアクセスするようになる。

TLB3内のTLBアクセス・アドレス6で指定される番地内には、論理アドレスオ0ないしオ4ビットとオ8ビットとに相当するページ・ナンバ(論理ページ・アドレス)が当該ページ・ナンバに対応する実ページ・アドレスと対にされて格納されている。該対を以下においてページ・アドレス対応と呼ぶ。

ホスト・マシンが論理アドレス2に対応する実アドレス5を決定するに当っては、次のように行なわれる。即ち、

- 1) 与えられた論理アドレス2のビットで構成されるTLBアクセス・アドレス6によって、TLB3がアクセスされる。
- 2) これによってTLB3から上述の如くページ・アドレス対応が読み出される。
- 3) 読み出されたページ・アドレス対応のうちのページ・ナンバはページ・ナンバ比較回路4に導びかれ、上記与えられた論理アドレス2で構成されるページ・ナンバ(オ0ないしオ4ビット

- 6) 上記処理5)によって得られた実アドレス5によって表示しない記憶装置がアクセスされる。

ホスト・マシンが与えられた論理アドレスを実アドレスに変換するに当って、上記の如き処理が実行されるが、特にエミュレート処理を行なう場合には高い頻度で該処理の途中にTLB3の内容を無効化し、以後の処理に当っては主記憶装置(表示せず)から改めてページ・アドレス対応をTLB内に転送することが行なわれる。該無効化処理に当っては、TLB消去命令時にTLB3内のすべてのページ・アドレス対応中のパリッド・ビットVをインパリッドしておき、後刻当該ページ・アドレス対応がアクセスされるとき比較回路部4が不一致を発し、新しいページ・アドレス対応をTLB上にもってくるようになる。この場合にも、TLB3内のすべてのページ・アドレス対応のパリッド・ビットVをインパリッドしてしまって、ホスト・マシンは新しい処理を実行することはできないために、上記インパリッド処理を高速で行なうことが望まれる。しかし、上記イ

とオ8ビットとで構成される)と比較される。そしてもしも不一致であれば、比較部4は不一致出力を発し、与えられた論理アドレス2を包含するページ・ナンバに当るページ・アドレス対応がTLB3内に存在していないか、あるいは存在していても当該対応内に用意されているパリッド・ビットVがインパリッド状態を指示していることを意味する。

- 4) 該不一致が発せられたときには、周知の如く、正当なページ・アドレス対応を表示しない主記憶装置からTLB3内に転送するようになる。
- 5) 上記比較部4が一致出力を発したとき、上記2)の処理によって読み出されたページ・アドレス対応が正当なものであることを意味している。このため、該読み出されたページ・アドレス対応のうちの実ページ・アドレスが実アドレス5の上位ビットにセットされる。そして実アドレス5の下位ビットに対し、与えられた論理アドレス2内のページ内アドレス(オ13ビットないしオ23ビット)がセットされる。

インパリッド処理に当って、ホスト・マシンは一般に、TLB3内の番地を1つ1つアクセスしてゆきパリッド・ビットVをインパリッドしてゆくようになる。

上記の点を考慮してオ1図を参照すると、TLBアクセス・アドレス6は、元論理アドレス1で見ると元論理アドレス1内のビットA0,A1,A2,A3,A4,A5,A6,A7に当っており、TLB3内のすべての番地(オ0番地ないしオ127番地)に対応していることが判る。このことから、上記TLB3の内容を無効化するに当っては、TLB3のすべての番地をアクセスしなければならない。しかし、それにも拘らず、TLB3内に格納されるページ・ナンバは論理アドレス2のオ0ビットないしオ4ビットとオ8ビットに対応していることから、ページ・ナンバを表すビットについてオ8ビットのみが有効内容をもっており他はすべて「0」をもっていることが知れる。即ち換言すると、TLB3内には単に2個分のページ・アドレス対応が存在しているだけであ

るに拘らず、上記インパリッド処理に当ってはTLB3内のオ0番地ないしオ127番地を順にアクセスしなければならない形となっている。

オ2図は上記の不都合を解決する本発明の一実施例構成を示す。図中の符号1ないし8はオ1図に対応し、9はTLB3内の制限された記憶領域を表わしている。

本発明の場合、元論理アドレス1を論理アドレス2に対応づけるに当って、附加される零ビット7をTLBアクセス・ビット領域8内に置くようになれる。オ2図図示の場合、元論理アドレス1におけるビットA0ないしA3を論理アドレス2におけるオ9ビットないしオ12ビットに、ビットA4ないしA7を論理アドレス2におけるオ0ビットないしオ3ビットに、附加される零ビット7を論理アドレス2におけるオ4ビットないしオ8ビットに対応づけるようにされる。これらの処理は、例えばマイクロプログラムによって実行せしめるようにすれば簡単に実行できる。

このため、TLBアクセス・アドレス6は図示

の如く「000A0,A1,A2,A3」の形をとることになる。即ち、TLB3の記憶領域は図示の場合オ0番地ないしオ15番地のみであるかの如くみなされることとなる。

ホスト・マシンが上記対応づけられた論理アドレス2にもとづいて実アドレス5を決定する処理を行なう態様は、オ1図図示の場合と全く同じと考えてよい。しかし、上述の如きTLB3の内容を無効化する処理に当っては、TLB3のオ0番地ないしオ15番地に対してアクセスを行なえば足りる。

以上説明した如く、本発明によれば上記エミュレーション処理を行なう如き場合、TLB3の番地が見掛け上制限された記憶領域のみをもつようにならざることになる。そのため、TLB3の内容を無効化する処理を行なう場合、きわめて簡単になる。また、元論理アドレス1に対応して、使用されるTLB3の記憶領域9が所定の範囲内に限られる。このため、複数個の元論理アドレスに対する処理を、TLB3上で互に独立せしめつ

つ並行して実行できる。

なお、上記の説明においてエミュレーション処理を例に挙げて説明したが、本発明はそれに限られないものではない。

4. 図面の簡単な説明

オ1図は従来のデータ処理方法を説明する説明図、オ2図は本発明のデータ処理方法を説明する説明図を示す。

図中1は元論理アドレス、2は論理アドレス、3は対応表(TLB)、4はページ・ナンバ比較部、5は実アドレス、6はTLBアクセス・ビット領域、7は附加される零ビット、8はTLBアクセス・ビット領域、9は制限された記憶領域を表わす。

特許出願人 富士通株式会社
代理人弁理士 森田 寛

